



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/964,763	09/27/2001	Arch D. Robison	042390.P11908	2972

8791 7590 06/24/2005

BLAKELY SOKOLOFF TAYLOR & ZAFMAN  
12400 WILSHIRE BOULEVARD  
SEVENTH FLOOR  
LOS ANGELES, CA 90025-1030

EXAMINER

YIGDALL, MICHAEL J

ART UNIT	PAPER NUMBER
----------	--------------

2192

DATE MAILED: 06/24/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

# Office Action Summary

Application No.

09/964,763

Applicant(s)

ROBISON, ARCH D.

Examiner

Michael J. Yigdall

Art Unit

2192

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

## Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

## Status

- 1) ☒ Responsive to communication(s) filed on 06 April 2005.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

## Disposition of Claims

- 4) ☒ Claim(s) 1,3-7,13,15-17,19-21 and 27-39 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1,3-7,13,15-17,19-21 and 27-39 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

## Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

## Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
  - ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

## Attachment(s)

- |  |   |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892)   | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)                                   | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152)             |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)<br>Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____  |

20

### **DETAILED ACTION**

1. Applicant's amendment and response filed on April 6, 2005 has been fully considered. Claims 1, 3-7, 13, 15-17, 19-21 and 27-39 are now pending.

### ***Response to Arguments***

2. Applicant's arguments have been fully considered but they are not persuasive.

Applicant contends that Archambault does not disclose or suggest determining whether a pointer is aliased with a restricted pointer, based on the base pointer and scope of each of the restricted pointer and the pointer, when the restricted pointer is out-of-scope relative to the pointer (Applicant's remarks, page 11, second paragraph). Specifically, Applicant contends that Archambault does not disclose using base pointers to determine whether an alias exists with a restricted pointer (Applicant's remarks, page 11, second paragraph), and that Archambault does not disclose using scope to determine whether an alias exists with a restricted pointer (Applicant's remarks, page 11, last paragraph).

However, Archambault discloses building a pointer alias graph and determining the alias set for each node in the graph, as acknowledged by Applicant. An "alias set" is a set of aliases, or the set of objects associated with a pointer variable (see, for example, column 2, lines 30-33). Each node in the graph represents a definition of a pointer or a use of a pointer, and the initial alias set for that pointer is set accordingly (see, for example, column 5, lines 8-15). As acknowledged by Applicant, the initial alias set for a node that represents a definition of a pointer is the "right hand side of the pointer variable assignment operation," or in other words, the address or l-value of the object of that pointer (see, for example, column 4, line 41). Also as

Art Unit: 2192

acknowledged by Applicant, the initial alias set for a node that represents a use of a pointer is the “value of the object at that execution point,” or in other words, the r-value of the object of that pointer (see, for example, column 4, lines 49-50).

Pointer variables are represented as l-values or r-values in the alias sets, and therefore the l-values and r-values may be considered “base pointers” for the pointer variables. In particular, the l-value of a pointer variable is a “base pointer” because it is the address to which the variable points, or in other words, it is the value of the pointer. Archambault expressly discloses determining aliases of pointer variables based on the l-values or base pointers (see, for example, column 6, lines 61-67, and column 8, lines 12-16).

Furthermore, Archambault teaches both intraprocedural alias analysis (see, for example, column 5, lines 4-17) and interprocedural alias analysis (see, for example, column 7, lines 27-33). Archambault discloses building a pointer alias graph for each function during intraprocedural analysis, and a universal pointer alias graph for the whole program during interprocedural analysis (see, for example, column 7, lines 34-45). Each function or procedure has a local scope, which is the scope for the pointers within that function or procedure (see, for example, column 5, lines 52-56). Thus, during intraprocedural analysis, Archambault determines aliases of pointer variables from the local scope of that procedure. During interprocedural analysis, Archambault determines aliases of pointer variables from more than one procedure, which is to say aliases of pointers from more than one scope. In fact, Archambault discloses that interprocedural analysis considers pointer variables of global or file scope (see, for example, column 5, lines 18-23). Therefore, Archambault discloses determining aliases of pointer variables based on the scope of the pointer variables.

Nonetheless, Applicant contends that Archambault does not disclose determining the scope of a restricted pointer in relation to the scope of another pointer, and using this scope information to determine whether an alias exists between the restricted pointer and the other pointer (Applicant's remarks, page 12, first paragraph).

However, Archambault discloses determining aliases of pointer variables based on the scope of the pointer variables, as noted above. Therefore, Archambault is aware of the scope of each pointer variable, and its scope relative to other pointer variables. Moreover, the interprocedural pointer analysis disclosed by Archambault could not be performed without identifying the scope of the pointers. As noted above, the scope of some pointer variables is local to a function or procedure (see, for example, column 5, lines 52-56), while the scope of other pointer variables is global (see, for example, column 5, lines 18-23).

As set forth in the previous Office action, the combination of Archambault, Blainey and Robison discloses determining whether a pointer is aliased with a restricted pointer when the restricted pointer is out-of-scope relative to the pointer. Therefore, in view of the above, the combination of Archambault, Blainey and Robison discloses determining whether a pointer is aliased with a restricted pointer, based on the base pointer and scope of each of the restricted pointer and the pointer, when the restricted pointer is out-of-scope relative to the pointer.

### *Claim Objections*

3. Claim 3 is objected to because it recites, "The method of claim 2." Claim 2 has been canceled by Applicant's amendment. Appropriate correction is required.

Art Unit: 2192.

4. Claim 15 is objected to because it recites, "The system of claim 14." Claim 14 has been canceled by Applicant's amendment. Appropriate correction is required.

5. Claim 19 is objected to because it recites, "The machine-readable medium of claim 18." Claim 18 has been canceled by Applicant's amendment. Appropriate correction is required.

***Claim Rejections - 35 USC § 103***

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

7. Claims 1, 3-7, 13, 15-17, 19-21 and 27-39 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 6,173,444 to Archambault (art of record, "Archambault") in view of U.S. Pat. No. 6,045,585 to Blainey (art of record, "Blainey") in view of "Restricted Pointers are Coming" by Robison (art of record, "Robison").

With respect to claim 1 (currently amended), Archambault discloses a method comprising:

(a) receiving a code segment having a plurality of instructions, the code segment having an outer scope and a number of inner scopes, wherein the plurality of instructions comprise a number of pointers (see, for example, column 5, lines 4-23, which shows receiving program code having a number of functions, i.e. a number of inner scopes within the global or outer scope of the program, and a plurality of instructions comprising a number of pointers).

Although Archambault discloses pointer variables in the C programming language (see, for example, column 4, lines 13-15), Archambault does not expressly disclose the limitation wherein at least one of the number of pointers is a restricted pointer.

However, Archambault discloses that the size of alias sets can be reduced with alias assertions made by the programmer (see, for example, column 2, lines 43-45), and that precise alias sets can be used to improve optimization (see, for example, column 3, lines 12-18).

Moreover, Blainey discloses that precise alias information can be determined based on language rules, language features and assertions made by the programmer (see, for example, column 2, lines 40-46). Blainey teaches a system for program optimization in a compiler (see, for example, column 3, lines 26-29).

Robison discloses an example of the language features suggested by Blainey, specifically restricted pointers and the use of the “restrict” keyword as an assertion made by the programmer in pointer declarations (see, for example, the “Restrict Qualifies Pointers” section). Robison further discloses that restricted pointers address problems associated with aliases and improve the performance of programs written in the C and C++ languages (see, for example, the “FORTRAN Envy” and “Conclusion” sections).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to supplement the alias analysis system of Archambault with language rules, language features and programmer assertions, so as to precisely determine aliases and reduce the size of alias sets, as taught by Blainey and as suggested by Archambault, including the language feature of restricted pointers disclosed by Robison, for the purpose of improving optimization and improving the performance of programs.

Therefore, Archambault in view of Blainey in view of Robison discloses the limitation wherein at least one of the number of pointers is a restricted pointer.

Archambault also discloses:

(b) determining a base pointer for each pointer of the number of pointers (see, for example, column 4, line 41 and column 5, lines 4-17, which shows determining an initial alias set for each pointer comprising the address or l-value of the pointer, i.e. the base pointer, from the pointer variable assignment operation);

(c) determining a scope of the at least one restricted pointer relative to scopes of each of the number of pointers in the code segment (see, for example, column 5, lines 52-56, which shows determining a local scope of the pointer relative to the scopes of pointers in other functions or procedures, and column 5, lines 18-23, which further shows pointers in the global scope of the code segment); and

(d) determining whether at least one pointer of the number of pointers is aliased with the at least one restricted pointer, based on the base pointer and scope of each of the restricted pointer and the at least one pointer, when the at least one restricted pointer is out-of-scope relative to the at least one pointer (see, for example, column 7, lines 27-45, which shows interprocedural alias analysis, i.e. determining whether a pointer is aliased with a pointer in another procedure, and column 5, lines 52-56, which shows that each function or procedure has a local scope, i.e. a pointer in one procedure is out-of-scope relative to a pointer in another procedure, and see, for example, column 6, lines 61-67, which shows determining aliases based on the l-value or base pointer and the local scope, and column 8, lines 12-16, which shows determining aliases based on the l-value or base pointer and the global scope).



With respect to claim 3 (currently amended), Archambault also discloses the limitation wherein the determining a base pointer for each pointer of the number of pointers comprises grouping pointers together upon determining that the pointers are copied to a pointer that is not a restricted pointer (see, for example, column 6, lines 16-46, which shows finding the union of alias sets and propagating transitive relationships to group pointers together, when appropriate, such as when pointers are copied to a non-restricted pointer, and column 6, lines 61-67, which further shows that the grouping results in alias sets comprising l-values or base pointers).

With respect to claim 4 (original), Archambault also discloses the limitation wherein there is no grouping of pointers when the pointers have distinct base pointers (see, for example, column 5, lines 31-41, which shows adding new nodes and alias sets to the pointer graph, i.e. not grouping the pointers into an alias set, when the base pointers are distinct and thus not already represented in the graph).

With respect to claim 5 (original), Archambault also discloses, for each instruction of the plurality of instructions that accesses a pointer, determining which at least one restricted pointer is within the scope of the pointer when the pointer is accessed (see, for example, column 5, lines 52-56, which shows determining the pointer variables accessed in the local scope).

With respect to claim 6 (previously presented), Archambault also discloses the limitation wherein the determining whether at least one pointer of the number of pointers is aliased with the at least one restricted pointer is based on the base pointer for each of the number of pointers (see,

Art Unit: 2192

for example, column 6, lines 61-67, and column 8, lines 12-16, which shows that alias sets are determined based on the l-values or base pointers).

With respect to claim 7 (previously presented), Archambault also discloses the limitation wherein the determining whether at least one pointer of the number of pointers is aliased with the at least one restricted pointer is based on, for each instruction of the plurality of instructions that accesses the pointer, which at least one restricted pointer is within the scope of the pointer, when the pointer is accessed (see, for example, column 5, lines 52-56, which shows determining the alias sets for all pointer variables accessed in the local scope).

With respect to claim 13 (currently amended), the claim recites a system that corresponds to the method recited in claim 1 (see the rejection of claim 1 above). Archambault also discloses a compiler coupled to a memory unit (see, for example, column 3, lines 12-18).

With respect to claim 15 (original), the limitations recited in the claim are analogous to those of claim 5 (see the rejection of claim 5 above).

With respect to claim 16 (previously presented), the limitations recited in the claim are analogous to those of claim 7 (see the rejection of claim 7 above).

With respect to claim 17 (currently amended), the claim recites a machine-readable medium that corresponds to the method recited in claim 1 (see the rejection of claim 1 above). Archambault also discloses a machine-readable medium that provides instructions to be executed by a machine (see, for example, column 3, lines 36-40).

With respect to claim 19 (original), the limitations recited in the claim are analogous to those of claim 5 (see the rejection of claim 5 above).

With respect to claim 20 (previously presented), the limitations recited in the claim are analogous to those of claim 6 (see the rejection of claim 6 above).

With respect to claim 21 (previously presented), the limitations recited in the claim are analogous to those of claim 7 (see the rejection of claim 7 above).

With respect to claim 27 (currently amended), Archambault also discloses the limitation wherein the at least one pointer of the number of pointers is aliased with the at least one restricted pointer for the same memory location if the at least two pointers have the same base pointer (see, for example, column 4, line 41, and column 2, lines 16-33, which shows that a pointer is aliased with another pointer for the same memory location if the pointers have the same l-value or base pointer).

With respect to claim 28 (currently amended), Archambault also discloses the limitation wherein if the at least two pointers do not have the same base pointers:

(a) determining whether each base pointer of the at least two pointers is a restricted pointer (see, for example, column 7, lines 13-19, which shows determining types of pointers, such as restricted pointers in view of Blainey in view of Robison).

Although Archambault discloses determining when pointers are in scope when the pointers are used, i.e. indirectly read or written (see, for example, column 5, lines 52-56), Archambault does not expressly disclose:

(b) if each base pointer is a restricted pointer, determining whether any of the base pointers is in scope when the other base pointers are indirectly read or written.

However, Robison discloses that the meaning and use of restricted pointers is associated with the scope of the pointers (see, for example, the “Restrict Qualifies Pointers” section).

As presented above, it would have been obvious to one of ordinary skill in the art at the time the invention was made to supplement the alias analysis system of Archambault with language rules, language features and programmer assertions, so as to precisely determine aliases and reduce the size of alias sets, as taught by Blainey and as suggested by Archambault, including the language feature of restricted pointers disclosed by Robison, for the purpose of improving optimization and improving the performance of programs.

Therefore, Archambault in view of Blainey in view of Robison discloses the limitation wherein if each base pointer is a restricted pointer, determining whether any of the base pointers is in scope when the other base pointers are indirectly read or written.

With respect to claim 29 (currently amended), Archambault also discloses the limitation wherein each of the at least two pointers are not aliases for the same memory location if each base pointer is in scope when the other base pointers are indirectly read or written (see, for example, column 2, lines 16-33, which shows that pointers may not be aliases for the same memory location if each base pointer is in scope when other base pointers are indirectly read or written).

With respect to claim 30 (currently amended), Archambault also discloses the limitation wherein each of the at least two pointers are aliases for the same memory location if at least one

Art Unit: 2192

base pointer is not in scope when the other base pointers are indirectly read or written (see, for example, column 7, lines 34-39, which shows the universal alias set for a pointer, showing that pointers may be aliases for the same memory location if at least one base pointer is not in scope when other base pointers are indirectly read or written).

With respect to claim 31 (currently amended), Archambault also discloses the limitation wherein each of the at least two pointers are aliases for the same memory location if any base pointer is parameter pointer and at least one base pointer is a restricted pointer that is not in scope when the other base pointers are indirectly read or written (see, for example, column 7, lines 16-33, which shows determining aliases based on types of pointers, such as argument or parameter pointers, and restricted pointers in view of Blainey in view of Robison, and see, for example, column 7, lines 34-39, which shows the universal alias set for a pointer, showing that pointers may be aliases for the same memory location if at least one base pointer is not in scope when other base pointers are indirectly read or written).

With respect to claims 32-36 (currently amended), the limitations recited in the claims are analogous to those of claims 27-31, respectively (see the rejections of claims 27-31 above).

With respect to claims 37-39 (new), the limitations recited in the claims are analogous to those of claims 27-29, respectively (see the rejections of claims 27-29 above).

*Conclusion*

8. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

9. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael J. Yigdall whose telephone number is (571) 272-3707. The examiner can normally be reached on Monday through Friday from 7:30am to 4:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.


Art Unit: 2192

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

MY

Michael J. Yigdall  
Examiner  
Art Unit 2192

mjy

  
**WEI Y. ZHEN**  
**PRIMARY EXAMINER**